



Administración de Versiones

REPOSITORIOS

Ariel Camilo Sánchez López | Programación 2 | 9/8/2021

Administración de Versiones

Los proyectos de software actuales cuentan con gran número de líneas de programación, al igual que gran cantidad de archivos, componentes y cambios. Por esa razón es necesario gestionar y controlar todos estos elementos para que se haga más simple la manera de entender la estructura a través del tiempo, es decir, entender las diferentes versiones con sus respectivos cambios, mejoras o soluciones.

Para realizar este control se hace uso de herramientas llamadas repositorios, estas herramientas permiten llevar un control temporal del proyecto en cuestión, se almacenan las diferentes versiones de los archivos teniendo en cuenta la fecha en qué se hizo, el autor y los respectivos cambios que se implementaron.

Los principales repositorios del mundo de la programación son los siguientes:

- GitHub
- Bitbucket
- CodePlex
- Codeplane
- Klin
- GitLab
- SourceForge

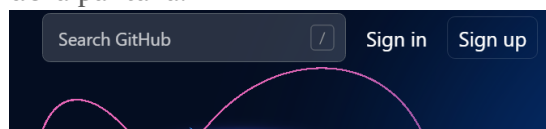
En esta ocasión haremos uso de GitHub para crear un repositorio y almacenar algunos proyectos.

GITHUB

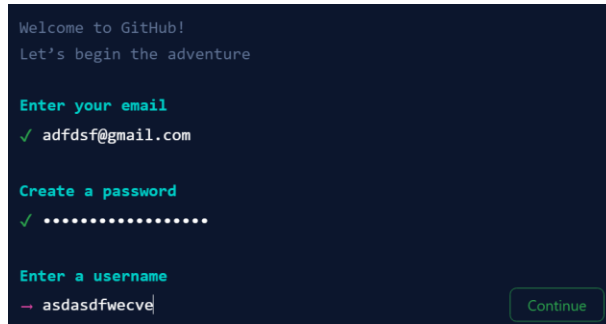
GitHub es uno de los repositorios más conocidos mundialmente, cuenta con una membresía gratuita donde los repositorios quedan en estado público, pero también permite que mediante una membresía premium los repositorios sean privados.

CREAR UN REPOSITORIO EN GITHUB

1. Registrarse en GitHub.
 - a. Entramos en la página oficial de GitHub, www.github.com.
 - b. Seleccionamos el botón Sign up ubicado en la parte derecha y superior de la pantalla.



- c. A continuación, introducimos el correo, contraseña y usuario.

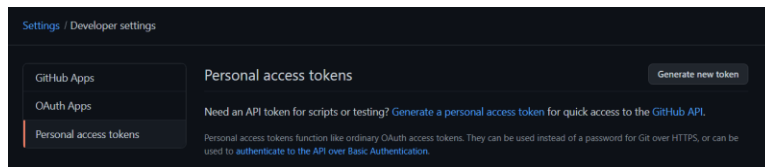
A screenshot of the GitHub registration form. It has a dark blue background with white text. The form includes fields for 'Enter your email' (with 'adfdsf@gmail.com' entered), 'Create a password' (with a masked password '.....'), and 'Enter a username' (with 'asdasdfwecve' entered). A green 'Continue' button is at the bottom right. The text 'Welcome to GitHub! Let's begin the adventure' is at the top.

- d. También se deberá resolver un puzzle para verificar el usuario y mediante un código enviado al correo que introduzcamos se verificará la existencia de este. Además, es necesario establecer un perfil del usuario en cuestión, pero se obviará por ahora.

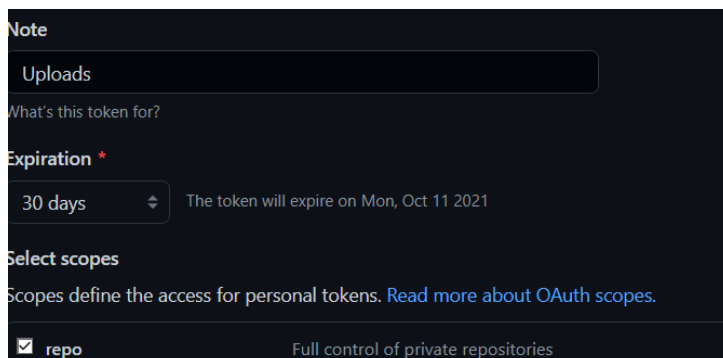
Para el primer repositorio se hará una guía propia de GitHub, en esta ocasión vamos a ver cómo crearlo desde cero.

2. Crear token de seguridad.

- a. En GitHub es necesario tener un token de seguridad para algunas tareas, por lo que crearemos uno. Dentro del sitio de GitHub nos dirigimos a “Settings/Developer Settings/Personal Access tokens” y presionamos “Generate new token”.

A screenshot of the GitHub 'Settings / Developer settings' page. The 'Personal access tokens' section is active. It shows a 'Generate new token' button and a description: 'Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API. Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.'

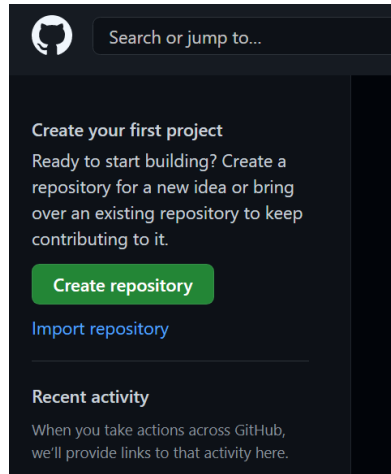
- b. Acá agregamos una descripción del uso del token y una fecha de expiración. Adicionalmente marcamos la casilla “repo” para poder tener control sobre los repositorios.

A screenshot of the GitHub 'Note' form for creating a personal access token. It includes a text input for 'Uploads', a 'What's this token for?' label, an 'Expiration' dropdown set to '30 days' (with a note 'The token will expire on Mon, Oct 11 2021'), and a 'Select scopes' section. Under 'Select scopes', the 'repo' checkbox is checked, with the description 'Full control of private repositories'.

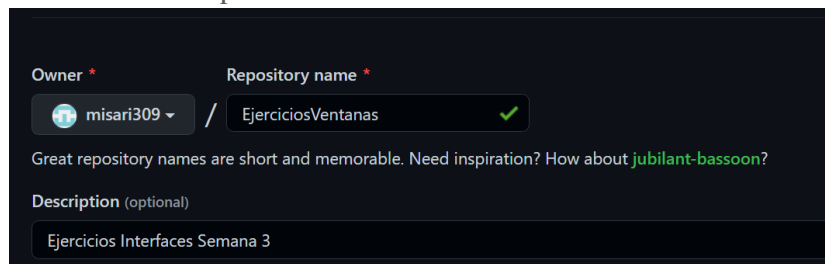
- c. Y creamos el token al final de la interfaz.

3. Crear repositorio en GitHub.

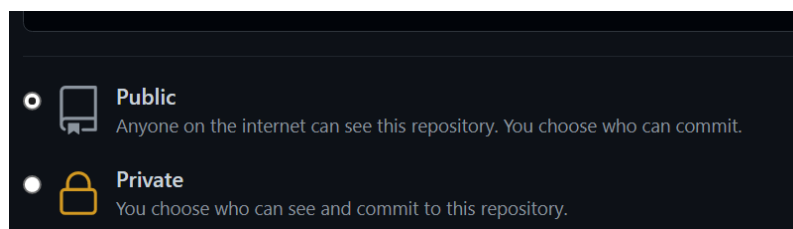
- a. Lo primero es situar nuestra vista en el menú izquierdo de la pantalla, allí encontraremos el botón 'Create Repository'.



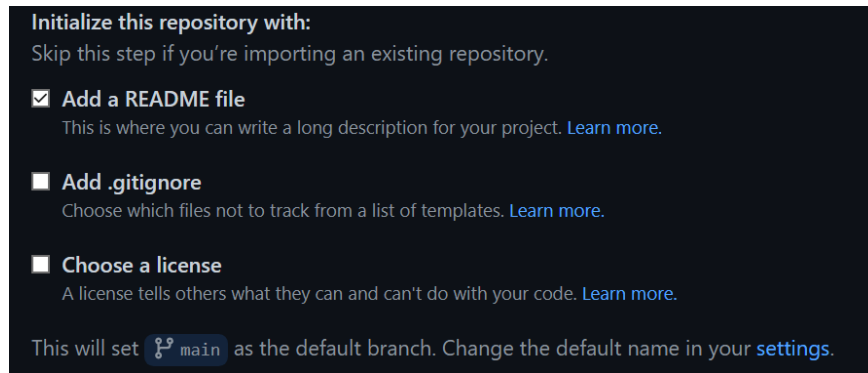
- b. Siguiendo a eso introduciremos el nombre de nuestro repositorio en la interfaz que se abre. También podemos agregar una descripción del contenido del repositorio.



- c. Elegimos la visibilidad del repositorio, recordar que solo se podrá poner en privado si contamos con membresía premium, en su defecto lo establecemos cómo público.



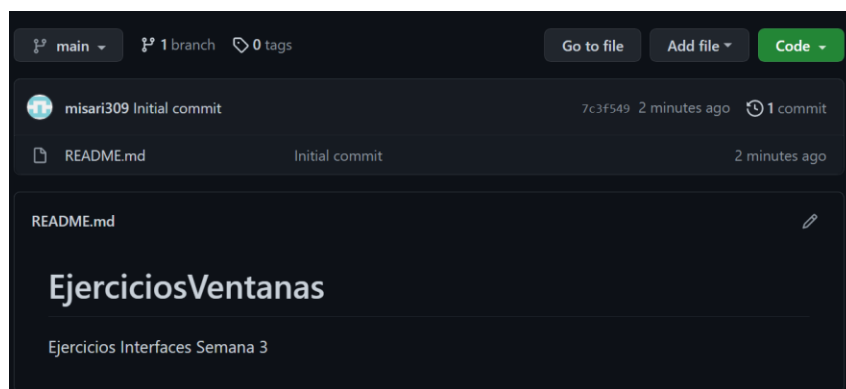
- d. Y para finalizar la creación del repositorio podemos agregar una licencia, al igual que un archivo README, donde estará una descripción extensa del repositorio.



e. Creamos el repositorio.

4. Agregar un proyecto desde la computadora.

- a. Nos encontraremos la siguiente interfaz, donde podemos agregar y visualizar los archivos que están dentro del repositorio. Para cargarlos es posible usar la línea de comandos o el propio sitio.



- b. Para agregar un proyecto desde la consola de comandos primero, naturalmente, debemos abrirla y ubicarnos en el directorio donde se encuentra el proyecto en cuestión.



- c. Antes de subir el archivo es necesario configurar la consola para que pueda subir esos archivos a GitHub, y se hace mediante los siguientes comandos. Este paso solo es necesario hacerlo la primera vez que se use este correo y usuario.

Para agregar nuestro correo.

```
ntanas$ git config --global user.email misari309@gmail.com|
```

Para agregar nuestro nombre de usuario.

```
s$ git config --global user.name misari309|
```

- d. A continuación, ejecutamos el comando “git init” para inicializar el repositorio en la carpeta. Y podemos usar el comando “git status” para verificar qué archivos se van a subir.

```
Semestre/EjerciciosVentanas$ git init  
erciciosTercerSemestre/EjerciciosVentanas/.git/  
Semestre/EjerciciosVentanas$ git status
```

- e. En este punto hay que agregar los archivos y para ello se usa el comando “git add .archivo”, pero si queremos agregarlos todos obviamos el nombre del archivo y ponemos un punto, “git add .”.

```
ntanas$ git add .
```

- f. Adicionalmente es necesario agregar un commit junto a la reciente acción, con la información de lo que sucedió, por ejemplo: “Se agregaron todos los archivos”. Y esto se hace mediante el siguiente comando.

```
Ventanas$ git commit -m "Se agregaron todos los archivos."|
```

- g. Por ahora solo hace falta especificar a qué rama van a pertenecer los archivos, dentro de este ejemplo pertenecen a la rama principal por lo que usaremos el siguiente comando.

```
nas$ git branch -M main|
```

- h. Ahora solo queda subirlo al propio repositorio en GitHub. Para ello usamos un comando junto a la dirección del repositorio de la siguiente manera.


```
s$ git remote add origin https://github.com/misari309/EjerciciosVentanas.git|
```

- i. Y para finalizar se confirma la vinculación entre lo que creamos dentro de la carpeta del proyecto y el repositorio en GitHub, para ello se usa el siguiente comando.

```
$ git push -u origin main
```

Acá tendremos que introducir nuestro username y el token que creamos previamente.

- j. Ya deberíamos tener nuestro proyecto en GitHub.

	misari309 Create README.md	26248ab · 14 minutes ago	🕒 2 commits
📁	build	Se agrega todo el documento	21 minutes ago
📁	nbproject	Se agrega todo el documento	21 minutes ago
📁	src/ejerciciosventanas	Se agrega todo el documento	21 minutes ago
📄	README.md	Create README.md	14 minutes ago
📄	build.xml	Se agrega todo el documento	21 minutes ago
📄	manifest.mf	Se agrega todo el documento	21 minutes ago

Enlace al repositorio: <https://github.com/misari309/EjerciciosVentanas.git>